

Estimating Causal Bounds from Observational Data With Neural Generative Models

Max Zhu* Srivathsan Amruth Sina Akbari Qingyuan Zhao

Department of Pure Mathematics and Mathematical Statistics
University of Cambridge

Abstract

Identifying causal bounds from discrete observational data is a fundamental problem in causal inference. While exact analytical bounds exist for certain graphs, this is not generally the case. In this paper, we introduce a framework for estimating such bounds using neural generative models together with a new extrapolation procedure to estimate these bounds more accurately. Across multiple causal inference settings, it produces bounds that are consistently closer to the true bounds than those obtained with existing general neural approaches. We validate the framework on a range of benchmark problems for which exact bounds are available.

1 Introduction

Estimating the effects of interventions from observational data is a central problem in causal inference. In many applications, however, the available observations and causal assumptions do not uniquely determine the causal quantity of interest. Hidden confounding, incomplete measurements, or limited experimental access may permit multiple structural causal models to generate the same observational distribution while implying different responses to an intervention. In such settings, reporting a single causal estimate can conceal uncertainty that persists even with complete knowledge of the observational distribution.

Partial identification provides a principled alternative to point estimation (Manski, 2003). Rather than selecting a single observationally compatible model, partial identification characterises the range of values attained by a causal query over all models satisfying the assumed causal structure and observational constraints. For example, the identified interval for an average treatment effect is obtained by minimising and maximising the effect over all structural models that reproduce the observed distribution. The resulting interval separates uncertainty arising from causal non-identifiability from ordinary uncertainty in parameter estimation.

For discrete causal models, causal bounds can be derived using linear, multilinear, or polynomial optimisation. Classical results provide sharp bounds in important special cases, such as instrumental-variable models (Balke and Pearl, 1997). More general approaches use logical relations between counterfactual events, canonical structural causal models, and automated polynomial-programming formulations (Song et al., 2025; Sachs et al., 2023; Boushehrian et al., 2026). These approaches offer valuable sharpness or certification guarantees, but the size of the resulting optimisation problem can increase rapidly with the number of variables, latent-variable configurations, and possible response functions. Furthermore, they have narrow applicability to general problems.

Recently, generative neural network approaches have been applied to this problem (Hu et al., 2021; Tan et al., 2024; Balazadeh Meresht et al., 2022). These attempt to find the extremal functions that satisfy the bound, providing both the bounds and functions that reach these bounds. While these are applicable to a wide range of problems, we find that their accuracy is limited.

This paper introduces a differentiable approach to causal partial identification for finite discrete structural causal models. Each structural mechanism is represented by a neural network whose output defines a

*Corresponding author: mz406@cam.ac.uk.

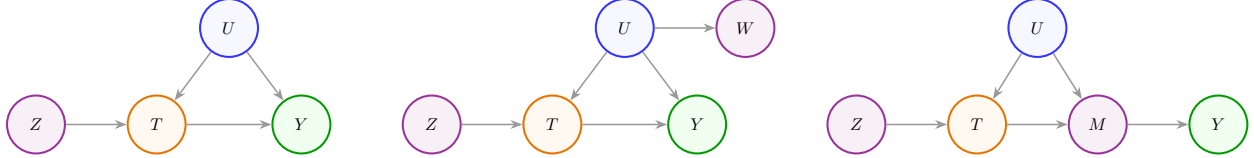


Figure 1: Causal graphs used in the simulation studies. Left: instrumental variable graph (IV). Centre: instrumental variable graph with a proxy variable (Proxy). Right: mediated instrumental variable graph (Mediated). U is exogenous and the other nodes are endogenous categorical variables.

conditional categorical distribution. The graph may contain explicit latent variables, which are marginalised to obtain the induced distribution over the observed variables.

We fit a neural network model by minimising the Energy Distance (Szekely, 2003) between the predictive model and the observational distribution, with a weighted causal loss. The Energy distance is a measure of distributional disagreement, minimised if the predicted and true distributions are identical. A novel extrapolation procedure, with theoretical justification, is introduced to estimate the extremal causal effects compatible with the observational distribution.

2 Related Work

2.1 Optimisation-based causal bounds

Partial identification characterises the range of causal quantities compatible with the observed distribution and the assumed causal structure (Manski, 2003). For discrete structural causal models, classical approaches represent latent structural mechanisms through deterministic response functions and optimise over their distributions. This yields linear programmes in important settings, including the instrumental-variable model with binary (Balke and Pearl, 1997) and general categorical variables (Song et al., 2025). Symbolic methods extend this construction to broader classes of causal graphs and queries and are implemented in tools such as `causaloptim` (Sachs et al., 2023, 2026). More general discrete problems can be expressed as polynomial programs (Zhang et al., 2022). Autobounds automatically constructs such programs and applies spatial branch-and-bound to obtain progressively tighter certified bounds (Duarte et al., 2024). Pruning and decomposition techniques can improve scalability for particular classes of graphs (Shridharan and Iyengar, 2023). These methods offer sharpness or certification guarantees in their supported settings, but their computational cost can grow rapidly with the number of variables and categories, and may fail to converge in more complex settings.

2.2 Neural approaches to partial identification

Neural causal models (Xia et al., 2021) provide an alternative to explicit response-function enumeration by representing structural mechanisms with flexible function approximators. Their connection to structural causal models and counterfactual reasoning has been studied theoretically, including representation and identifiability properties (Xia et al., 2022). Bounded Counterfactual Causal Effects (BCCE) parametrises structural mechanisms and latent variables using implicit generative models (Goodfellow et al., 2014) and employs adversarial training to match the observational distribution while optimising the causal effect (Hu et al., 2021). Related approaches impose distributional constraints using Wasserstein or Sinkhorn discrepancies and optimise them through Lagrangian procedures (Balazadeh Meresht et al., 2022; Tan et al., 2024). NeuralPartialID further establishes approximation and consistency results for continuous problems under regularity conditions such as Lipschitz continuity (Tan et al., 2024). Stochastic causal programming and implicit-generative approaches also address partial identification with continuous or multivariate treatments (Balazadeh Meresht et al., 2022; Padh et al., 2023). These methods broaden the range of causal models that can be considered.

2.3 Relation to our method

Our method builds on the neural causal partial-identification approaches. However, there are two key differences. Firstly, rather than comparing generated and observed samples through a learnt discriminator or an optimal-transport subproblem, we directly compute the probability assigned to every observed state and evaluate a closed-form energy distance between the resulting probability vectors. This avoids adversarial training and stochastic sampling of categorical model outputs.

The main distinction is our treatment of the exact-fit limit. Existing neural approaches attempt to optimise the causal query subject to, or jointly with, an observational-distribution constraint. In practice, this optimisation becomes difficult when the distributional discrepancy is required to be extremely small. We instead solve a signed family of stable penalised problems and extrapolate the causal contrast to the zero-penalty limit. Our perturbation analysis shows that, within a locally regular regime, the causal contrast varies linearly with the penalty parameter while the energy-distance loss varies quadratically. This provides both a principled extrapolation rule and a diagnostic for selecting the range over which extrapolation is valid.

3 Preliminaries

We introduce the causal and probabilistic notation used throughout. Random variables are denoted by upper case letters (e.g., U, X), their realisations by lower case letters (e.g., u, x), and vectors/sets of variables by bold letters (e.g., $\mathbf{U}, \mathbf{X}$). For a vector $\mathbf{X} = (X_1, \dots, X_n)$, where each X_i takes values in the finite set $\mathcal{X}_i$, the joint state space is

$$\mathcal{S} = \prod_{i=1}^n \mathcal{X}_i = \{\mathbf{s}_1, \dots, \mathbf{s}_K\}, \quad (1)$$

where $K = \prod_{i=1}^n |\mathcal{X}_i|$. A probability distribution on $\mathcal{S}$ can be represented by an element of the simplex

$$\Delta^{K-1} = \left\{ p \in \mathbb{R}_{\geq 0}^K : \sum_{k=1}^K p_k = 1 \right\}.$$

3.1 Structural equation models and causal graphs

A *structural equation model* (SEM) (Pearl, 2009) is a tuple

$$M = (\mathbf{U}, \mathbf{X}, \mathbf{F}, P_{\mathbf{U}}),$$

where $\mathbf{U} = (U_1, \dots, U_m)$ is a set of independent *exogenous* variables, $P_{\mathbf{U}}$ is their probability distribution, $\mathbf{X} = (X_1, \dots, X_n)$ is a set of *endogenous* variables, and $\mathbf{F} = (f_1, \dots, f_n)$ is a set of structural functions. Each endogenous variable is generated according to

$$X_i = f_i(\text{Pa}_{\mathbf{X}}(i), \text{Pa}_{\mathbf{U}}(i)),$$

where $\text{Pa}_{\mathbf{X}}(i) \subseteq \mathbf{X} \setminus \{X_i\}$ and $\text{Pa}_{\mathbf{U}}(i) \subseteq \mathbf{U}$ denote the set of endogenous and exogenous parents of X_i , respectively.

We assume that the directed relations between endogenous variables are acyclic. Consequently, the structural assignments can be evaluated in topological order. This induces a distribution on the endogenous variables which we denote by $P_M(\mathbf{X})$. The SEM can be represented by an acyclic directed graph (DAG) with vertices $\mathbf{X} \cup \mathbf{U}$, with an edge $X_i \rightarrow X_j$ whenever $X_i \in \text{Pa}_{\mathbf{X}}(j)$, and an edge $U_i \rightarrow X_j$ whenever $U_i \in \text{Pa}_{\mathbf{U}}(j)$. We denote this graph by $\mathcal{G}$.

Throughout this work, we consider a setting where $\mathbf{X}$ consists of discrete variables and $\mathbf{U}$ follows an unknown distribution that may be continuous or discrete. Sampling $\mathbf{U} \sim P_{\mathbf{U}}$ and evaluating the structural equations in topological order induces an observational distribution on $\mathcal{S}$.

3.2 Causal estimands and partial identification

A *do-intervention* on a variable ($X_i \in \mathbf{X}$), setting it to a value x , is denoted by $\text{do}(X_i = x)$. It modifies the structural equation model by replacing the structural equation for X_i ,

$$X_i = f_i(\text{Pa}_X(i), \text{Pa}_U(i)),$$

with the constant assignment

$$X_i := x.$$

All other structural equations remain unchanged. The intervention removes the causal influence of the original parents. This corresponds to removing all incoming edges to X_i in the causal DAG representing the model. We denote the resulting interventional distribution as

$$P_M(\mathbf{X} \setminus \{X\} \mid \text{do}(X = x)).$$

Let a denote a scalar causal estimand, and $a(M)$ denote this estimand evaluated in model M . For instance, $a(M)$ can denote the average treatment effect (ATE) of a binary treatment X on an outcome Y :

$$\text{ATE}_{X \rightarrow Y}(M) := \mathbb{E}_M[Y \mid \text{do}(X = 1)] - \mathbb{E}_M[Y \mid \text{do}(X = 0)],$$

which describes the expected causal difference on Y between applying a positive intervention (1) or applying a negative intervention (0).

Given a causal graph $\mathcal{G}$ and the observed distribution P_{obs} of the variables $\mathbf{X}$, the observational equivalence class is

$$\mathcal{M}(\mathcal{G}, P_{\text{obs}}) = \{M : M \text{ is compatible with } \mathcal{G} \text{ and } P_M(\mathbf{X}) = P_{\text{obs}}\}, \quad (2)$$

that is, the set of SEMs where parent sets agree with $\mathcal{G}$, and the induced distribution over $\mathbf{X}$ agrees with P_{obs} . The identified set of a causal estimand a given $\mathcal{G}$ and P_{obs} is then

$$\mathcal{I}_a(\mathcal{G}, P_{\text{obs}}) = \{a(M) : M \in \mathcal{M}(\mathcal{G}, P_{\text{obs}})\}.$$

When this set is a singleton, we say a is *point-identified*. Otherwise, it is only *partially identified*, and its (sharp) lower and upper bounds are

$$\underline{a} = \inf_{M \in \mathcal{M}(\mathcal{G}, P_{\text{obs}})} a(M), \quad \bar{a} = \sup_{M \in \mathcal{M}(\mathcal{G}, P_{\text{obs}})} a(M). \quad (3)$$

When the extrema are attained, the infimum and supremum may be replaced by a minimum and maximum. These bounds describe uncertainty due to causal non-identifiability even when P_{obs} is known exactly.

For the rest of this work, we focus on the ATE, although our method is applicable to arbitrary causal estimands.

3.3 Energy distance

Let P and Q be probability distributions on $\mathbb{R}^d$. Let $\mathbf{X}, \mathbf{X}' \stackrel{\text{iid}}{\sim} P$ and $\mathbf{Y}, \mathbf{Y}' \stackrel{\text{iid}}{\sim} Q$, with all random variables mutually independent. For $0 < \beta < 2$, we use the following convention for the energy distance between P and Q (Rizzo and Székely, 2016).

$$\text{ED}_\beta(P, Q) = 2 \mathbb{E} \|\mathbf{X} - \mathbf{Y}\|_2^\beta - \mathbb{E} \|\mathbf{X} - \mathbf{X}'\|_2^\beta - \mathbb{E} \|\mathbf{Y} - \mathbf{Y}'\|_2^\beta. \quad (4)$$

The energy distance measures distributional discrepancy through expected pairwise distances and therefore does not require a probability density evaluation. Because it can be estimated using only samples and is differentiable, it can also serve as a loss function for training generative models without requiring access to the model likelihood.

In the finite discrete case considered in this paper, define the pairwise distance matrix

$$\mathbf{D}_{k\ell} = \|\mathbf{s}_k - \mathbf{s}_\ell\|_2^\beta, \quad 0 < \beta < 2.$$

Proposition 1 (Finite-support energy distance). *Let $\mathbf{p}, \mathbf{q} \in \Delta^{K-1}$ be the probability vectors on $\mathcal{S}$. Then*

$$\text{ED}_\beta(\mathbf{p}, \mathbf{q}) = 2\mathbf{p}^\top \mathbf{D}\mathbf{q} - \mathbf{p}^\top \mathbf{D}\mathbf{p} - \mathbf{q}^\top \mathbf{D}\mathbf{q} = -(\mathbf{p} - \mathbf{q})^\top \mathbf{D}(\mathbf{p} - \mathbf{q}). \quad (5)$$

The full proof is given in Appendix C. In short, since we have finite support, every pairwise distance can be represented as an entry in $\mathbf{D}$, which leads to the simplified weighted probability sum.

Since $\mathbf{p} - \mathbf{q}$ has zero sum and $\mathbf{D}$ is strictly conditionally negative definite for distinct support points and $0 < \beta < 2$ (Schoenberg, 1938), this quantity is nonnegative and vanishes only when $\mathbf{p} = \mathbf{q}$.

4 Method

We describe our method for estimating causal estimands.

4.1 Neural parametrisation of discrete models

Recall from Section 3 that $\mathbf{X} = (X_1, \dots, X_n)$ denotes the endogenous variables and that X_i takes values in the finite state space $\mathcal{X}_i$. Define $k_i = |\mathcal{X}_i|$, and index the elements of $\mathcal{X}_i$ by $j \in \{0, \dots, k_i - 1\}$.

For each endogenous variable X_i , we introduce a neural network (NN) with parameters θ_i . Given $\mathbf{x}_{\text{Pa}\mathbf{X}(i)}$ and $\mathbf{u}_{\text{Pa}\mathbf{U}(i)}$, the values of endogenous and exogenous parents of X_i , respectively, the neural network produces a categorical probability vector

$$\boldsymbol{\pi}_i(\mathbf{x}_{\text{Pa}\mathbf{X}(i)}, \mathbf{u}_{\text{Pa}\mathbf{U}(i)}; \theta_i) = \text{softmax}(g_i(\mathbf{x}_{\text{Pa}\mathbf{X}(i)}, \mathbf{u}_{\text{Pa}\mathbf{U}(i)}; \theta_i)) \in \Delta^{k_i-1}, \quad (6)$$

where $g_i(\cdot; \theta_i)$ is the NN corresponding to X_i parametrised with θ_i , and a Softmax activation is applied to the output layer to ensure that the vector $\boldsymbol{\pi}$ lies in the probability simplex. We write

$$\pi_{i,j}(\mathbf{x}_{\text{Pa}\mathbf{X}(i)}, \mathbf{u}_{\text{Pa}\mathbf{U}(i)}; \theta_i)$$

for the probability of X_i taking value j . A separate network is used for each endogenous variable, and we collect all parameters as $\theta = (\theta_1, \dots, \theta_n)$.

Latent variables are modelled as i.i.d uniform random variables,

$$U_i \sim U(\mathbf{0}, \mathbf{1}).$$

This does not restrict our method to Uniform latent variables, since deep NNs can arbitrarily transform the latents to approximate the underlying latents (Yarotsky, 2018).

The parametrised mechanisms together with the exogenous distribution, define a neural SEM M_θ . Under this parametrisation, the observational and interventional distributions under M_θ can be computed as follows.

Observational distribution. For a subset $\mathbf{V} \subseteq \mathbf{X}$ of endogenous variables, let $\mathbf{s}_{k,\mathbf{V}}$ denote the values of $\mathbf{V}$ in state $\mathbf{s}_k$; see Equation (1). Conditional on a realisation $\mathbf{u}$ of the exogenous variables, the probability of $\mathbf{s}_k$ under the model M_θ factorises according to

$$p_\theta(\mathbf{s}_k \mid \mathbf{u}) = \prod_{i=1}^n \pi_{i,\mathbf{s}_{k_i}}(\mathbf{s}_{k,\text{Pa}\mathbf{X}(i)}, \mathbf{u}_{\text{Pa}\mathbf{U}(i)}; \theta_i). \quad (7)$$

Define the corresponding conditional probability vector

$$\mathbf{p}_\theta(\mathbf{u}) = (p_\theta(\mathbf{s}_1 \mid \mathbf{u}), \dots, p_\theta(\mathbf{s}_K \mid \mathbf{u}))^\top \in \Delta^{K-1}.$$

Marginalising the exogenous variables gives the observational distribution induced by M_θ :

$$\mathbf{p}_\theta = \mathbb{E}_{\mathbf{U}}[\mathbf{p}_\theta(\mathbf{U})] = (p_\theta(\mathbf{s}_1), \dots, p_\theta(\mathbf{s}_K))^\top \in \Delta^{K-1}. \quad (8)$$

This formulation gives a differentiable way to train our discrete model since Equation (7) is a product of predicted probabilities and for each exogenous realisation $\mathbf{u}$, the complete vector $\mathbf{p}_\theta(\mathbf{u})$ is computed by

enumerating the finite endogenous state space. We therefore do not sample categorical endogenous variables and do not require a Gumbel-Softmax (Jang et al., 2017) approximation. Furthermore, unlike policy-gradient, we do not need to stochastically sample over the categorical predictions.

The node networks also need not be evaluated separately for every complete state. Conditional independencies make computing the full vector $\mathbf{p}_\theta$ significantly faster. For each node X_i , outputs can be cached for each distinct configuration of its parents. Consequently, the number of node-network evaluations per exogenous draw is

$$\sum_{i=1}^n \prod_{j \in \text{Pa}_{\mathbf{X}}(i)} |\mathcal{X}_j|,$$

where the product is one for a node without endogenous parents. For the “straight line” graph where nodes form a simple chain, the compute complexity is $\mathcal{O}(nk)$. Further implementation details are provided in Appendix A

Interventional distribution. The same representation permits direct calculation of interventional distributions. Consider an intervention $do(X_t = x)$. Conditional on $\mathbf{U} = \mathbf{u}$, a state $\mathbf{s}_k$ is assigned zero probability unless $\mathbf{s}_{k_t} = x$, i.e., it is a compatible state. For compatible states, the factor associated with the intervened variable is removed:

$$p_\theta(\mathbf{s}_k \mid do(X_t = x), \mathbf{u}) = \mathbf{1}\{\mathbf{s}_{k_t} = x\} \prod_{\substack{i=1 \\ i \neq t}}^n \pi_{i, \mathbf{s}_{k_i}}(\mathbf{s}_{k, \text{Pa}_{\mathbf{X}}(i)}, \mathbf{u}_{\text{Pa}_{\mathbf{U}}(i)}; \theta_i).$$

This is the truncated factorization associated with an intervention, rather than ordinary conditioning on $X_t = x$. Marginal interventional probabilities $p_\theta(\mathbf{s}_k \mid do(X_t = x))$ are computed similarly to Equation (8). If $Y = X_r$ is the outcome variable, then

$$\mathbb{E}_\theta[Y \mid do(X_t = x)] = \sum_{k=1}^K \mathbf{s}_{k_r} p_\theta(\mathbf{s}_k \mid do(X_t = x)),$$

where $\mathbb{E}_\theta$ denotes expectation under the model M_θ with slight abuse of notation.

4.2 Empirical energy-distance loss

Here, we explain how the Energy Distance (Equation (5)) is estimated as a loss function for our training procedure. Let

$$\mathcal{D}_N = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}\}$$

be an observational dataset of N independent samples from P_{obs} , with each $\mathbf{x}^{(\ell)} \in \mathcal{S}$. We represent its empirical distribution by the probability vector

$$\hat{\mathbf{q}} = (\hat{q}_1, \dots, \hat{q}_K)^\top \in \Delta^{K-1},$$

where

$$\hat{q}_k = \frac{1}{N} \sum_{\ell=1}^N \mathbf{1}\{\mathbf{x}^{(\ell)} = \mathbf{s}_k\}, \quad k = 1, \dots, K. \quad (9)$$

Thus, $\hat{q}$ is the empirical analogue of the observational distribution P_{obs} introduced in Section 3.2. Further, for a neural network model M_θ , the expectation in Equation (8) is approximated using B independent exogenous draws $\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(B)}$:

$$\hat{\mathbf{p}}_\theta = \frac{1}{B} \sum_{b=1}^B \mathbf{p}_\theta(\mathbf{u}^{(b)}).$$

Directly substituting $\widehat{\mathbf{p}}_\theta$ into the quadratic term $\mathbf{p}^\top \mathbf{D} \mathbf{p}$ in Equation (5) is biased because the same Monte Carlo sample then appears in both factors (Gneiting and Raftery, 2007), introducing diagonal self-pair terms. We instead use a differentiable U-statistic. An unbiased estimator of $\mathbf{p}_\theta^\top \mathbf{D} \mathbf{p}_\theta$ is

$$\begin{aligned}\widehat{C}_\theta &= \frac{1}{B(B-1)} \sum_{\substack{b, b'=1 \\ b \neq b'}}^B \left(\mathbf{p}_\theta(\mathbf{u}^{(b)}) \right)^\top \mathbf{D} \mathbf{p}_\theta(\mathbf{u}^{(b')}) \\ &= \frac{B}{B-1} \widehat{\mathbf{p}}_\theta^\top \mathbf{D} \widehat{\mathbf{p}}_\theta - \frac{1}{B(B-1)} \sum_{b=1}^B \left(\mathbf{p}_\theta(\mathbf{u}^{(b)}) \right)^\top \mathbf{D} \mathbf{p}_\theta(\mathbf{u}^{(b)}).\end{aligned}$$

The empirical loss used for optimization is therefore

$$\widehat{\mathcal{L}}_{\text{ED}}(\theta; \beta) := \text{ED}_\beta(\widehat{\mathbf{p}}_\theta; \widehat{\mathbf{q}}) = 2\widehat{\mathbf{p}}_\theta^\top \mathbf{D} \widehat{\mathbf{q}} - \widehat{C}_\theta - \widehat{\mathbf{q}}^\top \mathbf{D} \widehat{\mathbf{q}}. \quad (10)$$

Conditional on $\widehat{\mathbf{q}}$, this is an unbiased estimator of $\widehat{\text{ED}}_\beta(\theta; \widehat{\mathbf{q}})$. Unlike the population Energy Distance, a finite-sample realization of the U-statistic may be slightly negative. This does not affect its differentiability or its use as an optimization objective. We use $\beta = 1$ in the rest of the manuscript and suppress the β parameter.

Every term in Equation (10) is differentiable with respect to θ , because the conditional probability vectors $\mathbf{p}_\theta(\mathbf{u}^{(b)})$ are computed directly from products of neural-network outputs. Further details of the estimator and its implementation are given in Appendix A.

4.3 Penalised optimisation formulation

Given the causal graph $\mathcal{G}$ and the empirical observational distribution $\widehat{\mathbf{q}}$ from Equation (9), the observational equivalence class of Equation (2) within the neural network model is

$$\Theta(\mathcal{G}, \widehat{\mathbf{q}}) = \{\theta \in \Theta_{\mathcal{G}} : \mathbf{p}_\theta = \widehat{\mathbf{q}}\} = \{\theta \in \Theta_{\mathcal{G}} : \mathcal{L}_{\text{ED}}(\theta; \widehat{\mathbf{q}}) = 0\}$$

by Proposition 1.

For brevity, we write $a(\theta) := a(M_\theta)$. The lower and upper causal bounds within the neural model family are therefore

$$\underline{a}_\Theta = \min_{\theta \in \Theta(\mathcal{G}, \widehat{\mathbf{q}})} a(\theta), \quad \bar{a}_\Theta = \max_{\theta \in \Theta(\mathcal{G}, \widehat{\mathbf{q}})} a(\theta).$$

These are neural-model, finite-sample analogues of the sharp population bounds defined in Equation (3). Rather than solving these constrained optimisation problems directly, for a scalar penalty parameter $\lambda \in \mathbb{R}$, we consider the penalised optimisation problems

$$\theta_\lambda = \arg \min_{\theta} J_\lambda(\theta), \quad (11)$$

where

$$J_\lambda(\theta) := \mathcal{L}_{\text{ED}}(\theta; \widehat{\mathbf{q}}) + \lambda a(\theta).$$

The sign of λ determines which extremum is favoured. For $\lambda > 0$, minimising Equation (11) favours smaller values of $a(M_\theta)$ and therefore traces the lower-bound. For $\lambda < 0$, it favours larger values and traces the upper-bound. As $|\lambda|$ decreases, observational fit receives increasing priority relative to the causal estimand. In the ideal case, the one-sided limits $\lambda \downarrow 0$ and $\lambda \uparrow 0$ approach models attaining $\underline{a}_\Theta$ and $\bar{a}_\Theta$, respectively.

In practice, however, optimisation becomes unreliable when $|\lambda|$ is extremely small (see Figure 2). We therefore do not take the estimate from the smallest available value of $|\lambda|$. Instead, we train at a series of stable nonzero values of λ and extrapolate the resulting causal estimates to $\lambda = 0$, as explained below.

4.4 Extrapolation

Let $I_+ = (0, \varepsilon)$, and $I_- = (-\varepsilon, 0)$ for some $\varepsilon > 0$.

Assumption 1 (smooth one-sided solution paths). *For each $s \in \{+, -\}$, there exists a map*

$$\theta_s : I_s \rightarrow \Theta_G$$

such that $\theta_s(\lambda)$ is a local minimiser of J_λ for every $\lambda \in I_s$. Moreover, the induced observational distribution $\mathbf{p}_{\theta_s(\lambda)}$ and causal estimand $a(\theta_s(\lambda))$ admit twice continuously differentiable one-sided extensions to $\lambda = 0$, with $\mathbf{p}_{\theta_s(0)} = \hat{\mathbf{q}}$.

Assumption 1 concerns the distributions and causal estimands represented by the neural networks; it does not require the raw parameter vectors $\theta_s(\lambda)$ to converge. Appendix D provides sufficient local KKT conditions under which Assumption 1 holds.

Theorem 1 (Local extrapolation). *Under Assumption 1, for each $s \in \{+, -\}$, there exist constants $c_s \in \mathbb{R}$ and $d_s \geq 0$ such that, as $\lambda \rightarrow 0$ within I_s ,*

$$a(\theta_s(\lambda)) = a(\theta_s(0)) + c_s \lambda + O(\lambda^2), \quad (12)$$

$$\mathcal{L}_{\text{ED}}(\theta_s(\lambda); \hat{\mathbf{q}}) = d_s \lambda^2 + O(|\lambda|^3). \quad (13)$$

Proof. Since $a(\theta_s(\lambda))$ is twice continuously differentiable at zero, its one-sided Taylor expansion gives

$$a(\theta_s(\lambda)) = a(\theta_s(0)) + a'(\theta_s(0))\lambda + O(\lambda^2),$$

where $a'(\theta_s(0)) := \frac{d}{d\lambda} a(\theta_s(\lambda))|_{\lambda=0}$. This proves Equation (12) with $c_s = a'(\theta_s(0))$.

Similarly, let

$$\mathbf{v}_s = \frac{d}{d\lambda} \mathbf{p}_{\theta_s(\lambda)}|_{\lambda=0}.$$

Using twice differentiability and Assumption 1,

$$\mathbf{p}_{\theta_s(\lambda)} - \hat{\mathbf{q}} = \mathbf{p}_{\theta_s(\lambda)} - \mathbf{p}_{\theta_s(0)} = \lambda \mathbf{v}_s + O(\lambda^2).$$

Substitution into the expression in Proposition 1 gives

$$\mathcal{L}_{\text{ED}}(\theta_s(\lambda); \hat{\mathbf{q}}) = -(\mathbf{p}_{\theta_s(\lambda)} - \hat{\mathbf{q}})^\top \mathbf{D} (\mathbf{p}_{\theta_s(\lambda)} - \hat{\mathbf{q}}) = -\lambda^2 \mathbf{v}_s^\top \mathbf{D} \mathbf{v}_s + O(|\lambda|^3).$$

Because $\mathbf{p}_{\theta_s(\lambda)}$ is a probability vector, $\mathbf{1}^\top \mathbf{v}_s = 0$. Conditional negative definiteness of $\mathbf{D}$ implies $\mathcal{L}_{\text{ED}}$ is nonnegative. $\square$

A more detailed analysis is given in Appendix D.

Assumption 2. *For every sufficiently small $\lambda \in I_s$, $\theta_s(\lambda)$ is a global minimiser of $J_\lambda(\theta) = \mathcal{L}_{\text{ED}}(\theta; \hat{\mathbf{q}}) + \lambda a(\theta)$.*

Corollary 1. *Under Assumptions 1 and 2,*

$$a(\theta_+(0)) = \underline{a}_\Theta(\hat{\mathbf{q}}), \quad a(\theta_-(0)) = \bar{a}_\Theta.$$

Consequently, the intercepts in Equation (12) for $s = +$ and $s = -$ equal the lower and upper neural causal bounds, respectively.

Proof. Let θ^* be any parameter such that $\mathbf{p}_{\theta^*} = \hat{\mathbf{q}}$, so that $\mathcal{L}_{\text{ED}}(\theta^*; \hat{\mathbf{q}}) = 0$. For $\lambda > 0$, global optimality gives

$$\mathcal{L}_{\text{ED}}(\theta_+(\lambda); \hat{\mathbf{q}}) + \lambda a(\theta_+(\lambda)) \leq \lambda a(\theta^*).$$

Since the energy-distance loss is nonnegative,

$$a(\theta_+(\lambda)) \leq a(\theta^*).$$

Taking $\lambda \downarrow 0$ and using $\mathbf{p}_{\theta_+(0)} = \hat{\mathbf{q}}$ shows that $a(\theta_+(0))$ is no larger than the estimand of any exact-fit model θ^* . Hence $a(\theta_+(0)) = \underline{a}_\Theta$. The argument for $\lambda < 0$ is analogous, with the inequality reversed. $\square$

Note that the branch derivatives $\mathbf{v}_s$ may differ for positive and negative λ . Finally, the Energy Distance is well defined throughout the simplex, unlike other (semi-)distances such as the Kullback–Leibler divergence, which becomes unbounded at the boundary and exhibits steep gradients and Hessians for small probabilities, potentially leading to more poorly conditioned fits.

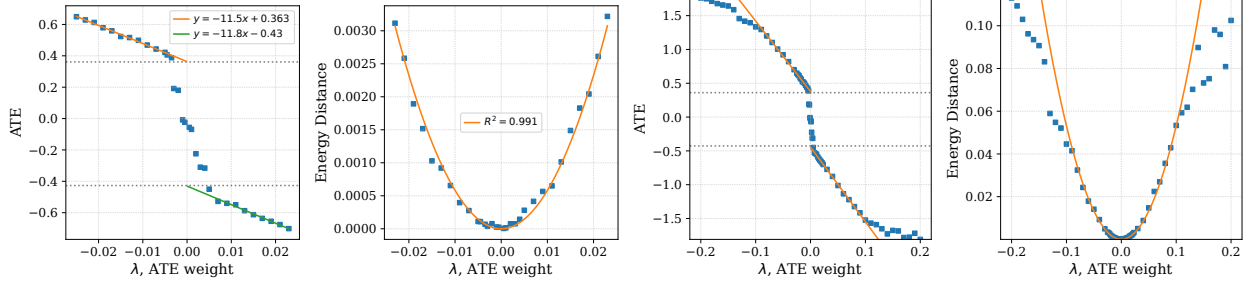


Figure 2: Model-predicted ATE and Energy Distance as λ varies. Left: values in the extrapolation regime, where fitted ATE intercepts recover the predicted bounds and Energy Distance follows a quadratic trend. The linear fit and R^2 goodness-of-fit values are shown for the plots. Right: values across a wider range of λ beyond the linear and quadratic regimes.

Dataset	Autobounds		BCCE		NeuralPartialID		Ours	
	Upper	Lower	Upper	Lower	Upper	Lower	Upper	Lower
IV-Binary	0	0	0.135 ± 0.012	0.087 ± 0.021	0.105 ± 0.026	0.159 ± 0.041	0.020 ± 0.005	0.011 ± 0.004
IV-Ternary	0	0	0.154 ± 0.043	0.125 ± 0.018	0.191 ± 0.065	0.162 ± 0.047	0.018 ± 0.011	0.010 ± 0.002
Proxy	$0.033 \pm 0.011^*$	$0.020 \pm 0.006^*$	0.093 ± 0.027	0.107 ± 0.029	0.067 ± 0.020	0.077 ± 0.032	0.032 ± 0.012	0.026 ± 0.009
Mediated	$0.136 \pm 0.038^*$	$0.171 \pm 0.041^*$	0.056 ± 0.020	0.037 ± 0.016	0.128 ± 0.053	0.099 ± 0.036	0.020 ± 0.009	0.021 ± 0.010

Table 1: Absolute error $\downarrow$ of the predicted upper and lower bounds for the ATE, compared with the true bounds. Standard errors are recorded over 10 datasets. (*) Autobounds does not fully converge and gives looser bounds than the true bound.

4.5 Training and bound estimation

Motivated by Theorem 1 and Corollary 1, we train models at several positive and negative values of λ and extrapolate their causal estimates to $\lambda = 0$. In practice, we minimise the objective

$$\hat{J}_\lambda(\theta) = \hat{\mathcal{L}}_{\text{ED}}(\theta; \hat{\mathbf{q}}) + \lambda a(\theta). \quad (14)$$

A series of different models with different positive and negative λ values are trained. We then regress the causal estimates on the λ values on the positive and negative branches. The intercept of the positive branch regression estimates the lower bound, while the one of the negative branch estimates the upper bound. *We do not solve the KKT conditions, rather we solve Equation (14) using SGD.*

The regression should use penalties within the local regime described by Theorem 1. However, this analysis does not consider if the optima can be found, only how the local region around them behaves. Empirically, using SGD on NNs we observe that convergence near $\lambda = 0$ is poor, but good convergence for small λ . Hence, we have a range of valid λ between the convergence and locally linear regime.

We select the regression points using the studentised-residual procedure given in algorithm 1 in the Appendix. The method begins with the largest penalty magnitudes and adds points closer to zero while they remain consistent with the fitted linear trend. The accompanying energy-distance values provide an additional diagnostic.

For all our experiments, we used 4 equally spaced points for positive and negative λ , $|\lambda| \in \{0.005, 0.010, 0.015, 0.020\}$. We found this range to be a good trade off between accuracy and training time, and is well within the linear regime. Further training details are given in Appendix A.

5 Simulation Studies

We evaluate our method against two neural baselines: NeuralPartialID (Tan et al., 2024) and Bounding Confounded Causal Effects (BCCE) (Hu et al., 2021), as well as Autobounds (Duarte et al., 2024).

Synthetic data is generated with four tasks derived from three causal graph structures: binary Instrumental Variable (IV), ternary IV, binary IV with a proxy variable (Proxy), and binary IV with a mediator (Mediated),

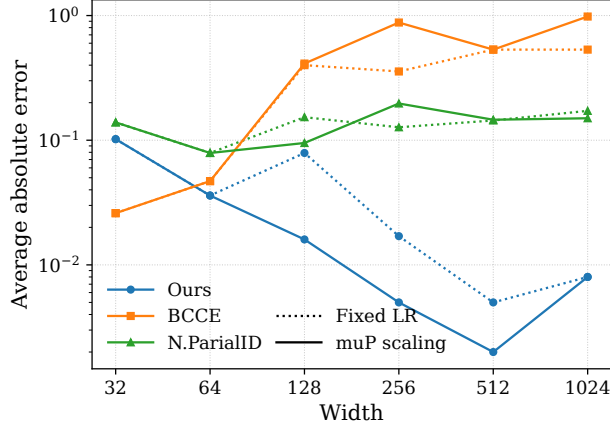


Figure 3: Absolute error between the analytical and predicted ATE bounds for our model, BCCE, and NeuralPartialID, averaged over the upper and lower bounds, over different neural network widths. Solid lines indicate a constant learning rate, while dashed lines indicate μP scaling.

shown in Figure 1. For each task, observational datasets are generated by randomly sampling from discrete SEMs whose mechanisms are defined by random Fourier functions. Details are given in Appendix F. For the IV and Proxy graphs, the ground truths are derived from the CausalOptim package (Sachs et al., 2026). However, CausalOptim and Autobounds are not able to handle the Mediated graph. For this type of graph, we derive the ground truth bounds in Appendix E.

This procedure gives a range of datasets that are guaranteed to satisfy the graphical constraints, together with corresponding ground-truth ATE bounds, without requiring explicit identification of the extremal functions. Each model was evaluated on 10 datasets for each graph type. Performance is measured by the absolute error in the estimated lower and upper ATE bounds, averaged across datasets.

5.1 Bounding accuracy

Table 1 gives the average absolute error in the ATE upper and lower bounds. Autobounds gives exact bounds for the IV Binary and IV Ternary problems, but does not fully converge for the Proxy graph, where the bounds are looser than the analytical bounds. Our method consistently gives much more accurate bounds than BCCE and NeuralPartialID on all graph types. On the Proxy graph, our method is comparable to Autobounds, while on the Mediated graph, our method significantly outperforms Autobounds.

5.2 Runtime

We measure the wall-clock runtime of the neural methods on a fixed ternary IV dataset, with identical hardware (Appendix A.5). Our method takes 330 seconds, while BCCE takes 84 seconds and NeuralPartialID takes 399 seconds. Our method is 20% faster than NeuralPartialID, though slower than BCCE. These runtimes should be interpreted together with the accuracy results, since the methods optimise different objectives and exhibit substantially different errors.

5.3 Behaviour of our approach

In order to visualise our method, on a fixed ternary IV dataset, we plot the ATE and $\mathcal{L}_{ED}$ for multiple values of λ in Figure 2. The shape of both the ATE and Energy Distance curves agrees with our theory. A linear extrapolation for the ATE gives the correct bounds, and the Energy Distance is well modelled by a quadratic curve. However, the model does not converge to the extremal bounds near $\lambda = 0$. Despite extensive tuning, this gap remains and necessitates our extrapolation method. The true ATE interval for this example is $[-0.427, 0.361]$. Linear extrapolation gives $[-0.430, 0.363]$, which closely matches the analytical bounds. In contrast, using the trained models directly without extrapolation gives approximately $[-0.450, 0.388]$.

Thus, the main improvement comes from estimating the zero-penalty intercept rather than taking the causal estimand from the model trained at the smallest available penalty. Figure 2 plots the ATE and $\mathcal{L}_{ED}$ for a wider range of λ . The ATE is linear in λ for around $0.005 < |\lambda| < 0.09$. At $\lambda = 0.09$, there is an abrupt change where both the ATE and energy distance stop following the linear and quadratic trend. Both changes are consistent with our theory; this corresponds to when a nodes predicted probability equals 0 or 1. There is a wide range of λ within the linear regime, with our selected λ inside it.

5.4 Analysis of the baselines

To investigate why the neural baselines underperform, we test each model using neural networks of varying widths. All other parameters are held constant, except for the learning rate, which is either scaled using μP scaling (Yang et al., 2022) or kept fixed. μP scaling promotes stable training across different widths by ensuring that the magnitudes of parameter updates and gradients remain stable. All models are trained on a fixed ternary IV SEM. Figure 3 shows the average absolute error between the predicted and analytical bounds.

Ideally, improving the width and hence representational capacity of the model improves performance. However, this is not observed for NeuralPartialID and BCCE, the accuracy decreases for larger NN widths. For BCCE, we attribute this to an instability in the GAN training procedure where the critic becomes too strong, preventing the generator from learning. Similarly, for NeuralPartialID, this is attributed to unstable parameter updates and hyperparameter sensitivity. Further analysis can be seen in Appendix G.

6 Conclusion

We introduced a neural approach to partial identification in finite discrete structural causal models. The method represents each structural mechanism by a neural categorical model and computes the induced observational and interventional distributions explicitly over the finite state space. Causal bounds are estimated by solving a signed family of energy-distance-penalised problems and extrapolating the resulting causal estimates to the zero-penalty limit. Our analysis shows that, along a smooth one-sided solution path, the causal estimand varies linearly with the penalty parameter to first order, while the energy-distance loss varies quadratically. The corresponding intercepts recover the lower and upper bounds within the neural model family. Across several simulations, our resulting extrapolated bounds are consistently more accurate than those of the evaluated neural baselines and remain competitive with analytical methods.

The approach is limited to discrete models and requires identifying a locally regular penalty range. Extending the approach to continuous variables and further development on the theory behind the validity of the linear extrapolation regime would broaden its practical scope. More generally, these results suggest that optimisation paths, rather than only their limiting iterates, can provide useful information for estimating causal bounds in partially identified models.

References

- Vahid Balazadeh Meresht, Vasilis Syrgkanis, and Rahul G Krishnan. Partial identification of treatment effects with implicit generative models. *Advances in Neural Information Processing Systems*, 35:22816–22829, 2022.
- Alexander Balke and Judea Pearl. Bounds on treatment effects from studies with imperfect compliance. *Journal of the American statistical Association*, 92(439):1171–1176, 1997.
- Arefe Boushehrian, Mohammad Reza Badri, Sina Akbari, and Negar Kiyavash. Fundamental limits and optimal methods for sharp analytical causal bounds in instrumental variable models. In *Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence*, volume 337 of *Proceedings of Machine Learning Research*, pages 689–744. PMLR, 2026.
- Marco Cuturi. Sinkhorn distances: lightspeed computation of optimal transport. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS’13, 2013.
- Guilherme Duarte, Noam Finkelstein, Dean Knox, Jonathan Mummolo, and Ilya Shpitser. An automated approach to causal inference in discrete settings. *Journal of the American Statistical Association*, 119(547):1778–1793, 2024.
- Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 2007.
- Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.
- Hinton. Rmsprop: Divide the gradient by a running average of its recent magnitude, 2012.
- Yaowei Hu, Yongkai Wu, Lu Zhang, and Xintao Wu. A generative adversarial framework for bounding confounded causal effects. In *AAAI Conference on Artificial Intelligence*, 2021.
- Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.
- Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2015.
- Charles F Manski. *Partial identification of probability distributions*. Springer, 2003.
- Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines, 2010.
- Kirtan Padh, Jakob Zeitler, David Watson, Matt Kusner, Ricardo Silva, and Niki Kilbertus. Stochastic causal programming for bounding treatment effects. In *Conference on Causal Learning and Reasoning*, pages 142–176. PMLR, 2023.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019.
- Judea Pearl. *Causality*. Cambridge University Press, 2009.
- Maria L. Rizzo and Gábor J. Székely. Energy distance. *WIREs Computational Statistics*, 8(1):27–38, 2016. doi: <https://doi.org/10.1002/wics.1375>.
- Michael C Sachs, Gustav Jonzon, Arvid Sjölander, and Erin E Gabriel. A general method for deriving tight symbolic bounds on causal effects. *Journal of Computational and Graphical Statistics*, 32(2):567–576, 2023.
- Michael C Sachs, Erin E Gabriel, Arvid Sjölander, and Gustav Jonzon. *causaloptim: An Interface to Specify Causal Graphs and Compute Bounds on Causal Effects*, 2026. URL <https://sachsmc.github.io/causaloptim/>.

- Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016.
1. J. Schoenberg. Metric spaces and positive definite functions, 1938.
- Madhumitha Shridharan and Garud Iyengar. Scalable computation of causal bounds. *Journal of Machine Learning Research*, 24(237):1–35, 2023.
- Yilin Song, F. Richard Guo, K. C. Gary Chan, and Thomas S. Richardson. The categorical instrumental variable model: Characterization, partial identification, and statistical inference, 2025.
- Gabor J. Szekely. E-statistics: The energy of statistical samples. *Bowling Green State University, Department of Mathematics and Statistics Technical Report*, 2003.
- J. Tan, J. Blanchet, and V. Syrgkanis. Consistency of Neural Causal Partial Identification. In *Advances in Neural Processing Systems 2024*, 2024.
- Kevin Xia, Kai-Zhan Lee, Yoshua Bengio, and Elias Bareinboim. The causal-neural connection: expressiveness, learnability, and inference. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, 2021.
- Kevin Xia, Yushu Pan, and Elias Bareinboim. Neural causal models for counterfactual identification and estimation. *arXiv preprint arXiv:2210.00035*, 2022.
- Greg Yang, Edward J. Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer. In *Advances in Neural Processing Systems 2021*, 2022.
- Dmitry Yarotsky. Optimal approximation of continuous functions by very deep relu networks, 2018.
- Junzhe Zhang, Jin Tian, and Elias Bareinboim. Partial counterfactual identification from observational and experimental data. In *Proceedings of the 39th International Conference on Machine Learning*, 2022.

Algorithm 1 Lower (Upper) ATE bound estimation via studentized-residual fit

Input: Data vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^L$; minimum points $k_{\min}$; threshold τ

Output: Intercept estimate $\hat{\beta}_0$

```
1: Sort  $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^L$  by  $\mathbf{x}_i$  descending (ascending), yielding  $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ .  
2:  $S \leftarrow \{0, \dots, k_{\min} - 1\}$  // initial set  
3: for  $i = k_{\min}, \dots, L - 1$  do  
4:    $T \leftarrow S \cup \{i\}$  // tentatively add next point  
5:   Fit OLS:  $\mathbf{y}_T \sim \beta_0 + \beta_1 \tilde{\mathbf{x}}_T$   
6:   Compute internally studentised residuals  $\mathbf{r}$ .  
7:   if  $|\mathbf{r}_{-1}| < \tau$  then  
8:      $S \leftarrow T$  // accept  
9:   else  
10:    break // reject and stop  
11:  end if  
12: end for  
13: Fit  $\tilde{\mathbf{y}}_S \sim \beta_0 + \beta_1 \tilde{\mathbf{x}}_S$  on final  $S$ .  
14: return  $\hat{\beta}_0$ 
```

A Implementation Details

Our model uses the assumed causal DAG to construct a neural structural causal model. As in the data-generating process, the nodes are evaluated in topological order. However, the unknown structural mechanisms are represented by neural networks rather than fixed random functions.

Let $X_1, \dots, X_m$ denote the discrete endogenous nodes and let $\mathbf{U}$ denote the exogenous variables. For every node X_i , the model specifies the categorical probabilities

$$\pi_{i,j}(\text{Pa}_X(i), \text{Pa}_U(i); \theta_i), \quad j \in \{0, \dots, k_i - 1\},$$

where $\text{Pa}_X(i)$ and $\text{Pa}_U(i)$ are the endogenous and exogenous parents of X_i , respectively. The exogenous variables are sampled as in the main model, $\mathbf{U} \sim U(\mathbf{0}, \mathbf{1})$.

For a node with k_i possible states, the values of its parents are concatenated into the input

$$\mathbf{h}_i^{(0)} = \text{concat}(\text{Pa}_X(i), \text{Pa}_U(i)).$$

This input is passed through a multilayer perceptron,

$$\begin{aligned} \mathbf{h}_i^{(\ell)} &= \rho \left(W_i^{(\ell)} \mathbf{h}_i^{(\ell-1)} + \mathbf{b}_i^{(\ell)} \right), \quad \ell = 1, \dots, L, \\ \mathbf{a}_i &= W_i^{(\text{out})} \mathbf{h}_i^{(L)} + \mathbf{b}_i^{(\text{out})}, \end{aligned}$$

where ρ is the ReLU activation function and $\mathbf{a}_i \in \mathbb{R}^{k_i}$ contains the output logits. The conditional probability of state j is obtained using a softmax:

$$\pi_{i,j}(\text{Pa}_X(i), \text{Pa}_U(i); \theta_i) = \frac{\exp(a_{i,j})}{\sum_{\ell=0}^{k_i-1} \exp(a_{i,\ell})}.$$

A separate neural network with parameters θ_i is used for each node.

A.1 Joint probability calculation

The model computes probabilities explicitly over all possible joint configurations of the discrete endogenous nodes. Let

$$S = \prod_{i=1}^m \{0, \dots, k_i - 1\}$$

be the Cartesian product of their state spaces. The number of configurations is therefore

$$K = \prod_{i=1}^m k_i.$$

We denote the c -th configuration by $\mathbf{s}_c \in \mathcal{S}$.

For a given realisation $\mathbf{u}$ of the exogenous variables, the probability of a configuration factorizes according to the DAG:

$$p_\theta(\mathbf{s}_c \mid \mathbf{u}) = \prod_{i=1}^m \pi_{i, s_{c,i}}(\text{Pa}_{\mathbf{s}_c}(i), \text{Pa}_{\mathbf{u}}(i); \theta_i). \quad (15)$$

Thus, each network returns a probability vector over the states of its node, and the entry corresponding to $s_{c,i}$ is selected for each global configuration. The selected conditional probabilities are then multiplied across nodes.

To avoid unnecessary network evaluations, the model identifies the unique configurations of a node's discrete parents. The node network is evaluated once for each unique parent configuration rather than once for every complete configuration in $\mathcal{S}$. Its outputs are then mapped back to the corresponding complete configurations. This optimization does not change the probability represented by Equation (15).

A.2 Marginal probabilities

The probabilities in Equation (15) are conditional on a realisation of the exogenous variables. The model distribution is obtained by integrating over $\mathbf{U}$:

$$p_\theta(\mathbf{s}) = \mathbb{E}_{\mathbf{U}}[p_\theta(\mathbf{s} \mid \mathbf{U})].$$

This expectation is approximated by Monte Carlo sampling. Given B independent exogenous realisations,

$$\hat{p}_\theta(\mathbf{s}) = \frac{1}{B} \sum_{b=1}^B p_\theta(\mathbf{s} \mid \mathbf{u}^{(b)}).$$

A.3 Interventional probabilities

Interventions are implemented by replacing the model assignment for the intervened node. Under the intervention $\text{do}(X_t = a)$, configurations for which $s_{c,t} \neq a$ are assigned probability zero. For compatible configurations, the conditional factor associated with X_t is omitted:

$$\begin{aligned} & p_\theta(\mathbf{s}_c \mid \text{do}(X_t = a), \mathbf{u}) \\ &= \mathbf{1}_{\{s_{c,t} = a\}} \prod_{i \in \{1, \dots, m\} \setminus \{t\}} \pi_{i, s_{c,i}}(\text{Pa}_{\mathbf{s}_c}(i), \text{Pa}_{\mathbf{u}}(i); \theta_i). \end{aligned}$$

This is the truncated factorization associated with an intervention, rather than ordinary conditioning on $X_t = a$.

For an outcome node Y , its expected value under treatment a is computed directly from the probabilities of the enumerated states:

$$\begin{aligned} & \mathbb{E}_\theta[Y \mid \text{do}(X_t = a)] \\ &= \mathbb{E}_{\mathbf{U}} \left[\sum_{c=1}^K s_{c,Y} p_\theta(\mathbf{s}_c \mid \text{do}(X_t = a), \mathbf{U}) \right]. \end{aligned}$$

The causal contrast used in the main objective is consequently

$$a(\theta) = \mathbb{E}_\theta[Y \mid \text{do}(X_t = 1)] - \mathbb{E}_\theta[Y \mid \text{do}(X_t = 0)].$$

A.4 Estimating Energy Distance

Proposition 1 gives the Energy Distance for a random discrete model. However, when estimating the Energy Distance from samples from finite samples, a modified formula is used (Rizzo and Székely, 2016). Specifically, the term involving the distance between predictions, $\mathbb{E}||x - x'||$ is modified. The term can be estimated with the unbiased U estimator

$$\begin{aligned}\mathbb{E}||x - x'|| &\approx \frac{1}{B^2} \sum_{i \neq j} \mathbf{p}_\theta^{i\top} \mathbf{D} \mathbf{p}_\theta^j \\ &= \frac{B}{B-1} \tilde{\mathbf{p}}_\theta \mathbf{D} \tilde{\mathbf{p}}_\theta - \frac{1}{B(B-1)} \sum_{i=1}^B \mathbf{p}_\theta^{i\top} \mathbf{D} \mathbf{p}_\theta^i\end{aligned}$$

where $\tilde{\mathbf{p}}$ is the mean predicted probability, or the biased V estimator

$$\mathbb{E}||x - x'|| \approx \tilde{\mathbf{p}} \mathbf{D} \tilde{\mathbf{p}}.$$

Using the V estimator means the Energy Distance is always positive, but biased positively. Alternatively, the U estimator gives an unbiased Energy Distance, but possibly negative empirical value. Since our method can handle negative distances, we use the U estimator. We note the difference between the two estimators is $\mathcal{O}(B^{-2})$ and makes very little difference for large batches.

A.5 Hyperparameters

We use deep neural networks for our model to parametrize Equation (6). The same model hyperparameters are used for all experiments. Our models have hidden width 1024 and depth 3. The input and output dimensions are determined based on the parent nodes and output categories. The ReLU activation function (Nair and Hinton, 2010) is used for the hidden layers, with the Softmax activation on the output to constrain the output to valid probabilities. Our model is trained with the Adam optimiser (Kingma and Ba, 2015) with a cosine decay schedule and maximum learning rate of 1e-4 and 24 epochs with 256 steps per epoch. The linear fit threshold was selected as $\tau = 1.25$. Our hyperparameters were tuned on a single Ternary IV graph, like the baselines. The random seed for each run is the same as the one used to generate the datasets, that is $\{0, \dots, 9\}$.

Finally, our method is implemented using PyTorch (Paszke et al., 2019) version 2.13. All experiments and timings (including baselines) are conducted using Ubuntu Linux 26.04. The hardware is as follows: GPU NVIDIA RTX 3090, CPU AMD RYZEN 5800X3D and 32GB DDR4 system memory.

B Baselines and Evaluation Metrics

For each dataset, we compute the observed distribution $\mathbf{q}$ using $B = 1000000$ samples. The baselines compute their predicted bounds using these observed probabilities.

The baselines are implemented using the official packages / code.

Autobounds (Duarte et al., 2024) uses the default configuration, used from their Docker container.

CausalOptim (Sachs et al., 2026) generates analytical bounds given a causal graph, given as a series of equations. For each graph type, we generate the analytical equations for the upper and lower bounds.

NeuralPartialID (Tan et al., 2024) uses the official implementation. We tuned the hyperparameters using a sequential grid search to find the best parameters. The range of parameters searched were: Width [16, 32, 64, 128, 256, 512, 1024], depth [2, 3, 4], learning rate [3e-4, 1e-3, 3e-3], lagrange learning rate [0.1, 0.3, 0.5, 0.7]. Compared to the original implementation, we settled on increasing the depth of the NNs from 2 to 3, increasing the Lagrange learning rate from 0.1 to 0.5 and increasing the width from 32 to 64 with a max training length of 500 epochs, which we found gives the best performance on a standard task.

BCCE (Hu et al., 2021) also uses the official implementation. We used the *Nonlinear* model class for our experiments so the models are general non-linear neural networks. The default hyperparameters were very unstable and yielded poor results, so the parameters were manually tuned. For the final hyperparameters, the width of both the generator and critic was increased from 16 to 64, and the depth of the generator was

increased to 3. The generator’s optimiser was switched from RMSProp (Hinton, 2012) to AdamW, while RMSProp was kept for the discriminator. The learning rate was increased from 3e-4 to 5e-2 for both the discriminator and generator. Finally, a stepwise learning rate decay schedule was added of 0.5 every 200 epochs. These changes significantly increased the accuracy. Increasing the training duration from the default 2000 epochs did not improve performance, but led to training instabilities so the duration was fixed at 500 epochs.

The hyperparameters for both BCCE and NeuralPartialID were configured on a Ternary IV graph sample.

C Finite-Support Energy Distance

This section proves the finite-support representation in Proposition 1. Let $\mathbf{p} = (p_1, \dots, p_K)^\top$ and $\mathbf{q} = (q_1, \dots, q_K)^\top$. Let $\mathbf{X}, \mathbf{X}' \sim \mathbf{p}$ and $\mathbf{Y}, \mathbf{Y}' \sim \mathbf{q}$ independently.

Proof of Proposition 1. Expanding the cross-distribution expectation gives

$$\begin{aligned}\mathbb{E}\|\mathbf{X} - \mathbf{Y}\|_2^\beta &= \sum_{k,\ell=1}^K P(\mathbf{X} = \mathbf{s}_k)P(\mathbf{Y} = \mathbf{s}_\ell)\|\mathbf{s}_k - \mathbf{s}_\ell\|_2^\beta \\ &= \sum_{k,\ell=1}^K p_k q_\ell \mathbf{D}_{k\ell} = \mathbf{p}^\top \mathbf{D} \mathbf{q}.\end{aligned}$$

Likewise,

$$\mathbb{E}\|\mathbf{X} - \mathbf{X}'\|_2^\beta = \sum_{k,\ell=1}^K p_k p_\ell \mathbf{D}_{k\ell} = \mathbf{p}^\top \mathbf{D} \mathbf{p},$$

and

$$\mathbb{E}\|\mathbf{Y} - \mathbf{Y}'\|_2^\beta = \sum_{k,\ell=1}^K q_k q_\ell \mathbf{D}_{k\ell} = \mathbf{q}^\top \mathbf{D} \mathbf{q}.$$

Substituting these terms into Equation (4) gives

$$\begin{aligned}\text{ED}_\beta(\mathbf{p}, \mathbf{q}) &= 2\mathbf{p}^\top \mathbf{D} \mathbf{q} - \mathbf{p}^\top \mathbf{D} \mathbf{p} - \mathbf{q}^\top \mathbf{D} \mathbf{q} \\ &= -(\mathbf{p} - \mathbf{q})^\top \mathbf{D} (\mathbf{p} - \mathbf{q}),\end{aligned}$$

where the final equality uses the symmetry of $\mathbf{D}$. □

By Schoenberg’s theorem (Schoenberg, 1938), $\mathbf{D}$ is strictly conditionally negative definite for distinct support points and $0 < \beta < 2$. Since $\mathbf{p} - \mathbf{q}$ has zero sum, the quadratic form above is nonnegative and is zero if and only if $\mathbf{p} = \mathbf{q}$.

D KKT Perturbation Derivation

The main text uses softmax outputs, so the raw neural parameters θ are unconstrained while the conditional probability vectors are constrained to the simplex implicitly. The KKT derivation below is written in ambient mechanism coordinates rather than in the raw neural coordinates. This makes the derivation applicable to the induced solution path without requiring the raw parameter vectors to converge or be unique.

D.1 Constrained mechanism formulation

Let η denote local coordinates for the conditional mechanism probabilities, and let $\boldsymbol{\pi}(\eta)$ collect all corresponding probability vectors. For each $s \in \{+, -\}$, we study a local-minimiser branch of the abstract constrained problem

$$\begin{aligned}\min_{\eta} \quad & \mathcal{L}_{\text{ED}}(\eta; \hat{\mathbf{q}}) + \lambda a(\eta) \\ \text{subject to} \quad & \mathbf{A}(\eta) = \mathbf{0}, \quad \mathbf{B}(\eta) \leq \mathbf{0}.\end{aligned}$$

where the simplex constraints are included in $\mathbf{A}$ and $\mathbf{B}$. For example, each conditional mechanism satisfies

$$\sum_j \pi_{i,j}(\eta) = 1, \quad -\pi_{i,j}(\eta) \leq 0.$$

The neural model in the main text is a smooth feasible map

$$\eta = \Phi(\theta), \quad \pi(\Phi(\theta)) = \text{softmax}(g(\theta)).$$

Thus, the explicit constraints are eliminated in the implementation, but the induced observational distribution and causal estimand lie in the same feasible mechanism space. At strictly positive probability vectors, the softmax map is locally onto the simplex tangent space because its Jacobian has rank $k - 1$. We therefore use the constrained coordinates η only for the local sensitivity calculation. The result transfers to the softmax model whenever Φ is locally expressive on the feasible tangent space. Softmax does not represent boundary probabilities exactly; boundary solutions are understood as limits of the interior analysis.

D.2 Local regularity assumptions

The following assumptions make the constrained KKT system a valid local description of the solution paths in Assumption 1. Fix a regular reference point (η_0, λ_0) with $\lambda_0 \neq 0$, and let $\mathcal{A} = \{r : B_r(\eta_0) = 0\}$ be the active inequality set.

First, $\mathcal{L}_{\text{ED}}(\eta; \hat{\mathbf{q}})$, $a(\eta)$, $\mathbf{A}(\eta)$, and $\mathbf{B}(\eta)$ are twice continuously differentiable near η_0 . The active set is fixed under the local perturbation, and the rows of

$$\mathbf{C}_0 = \begin{bmatrix} \nabla_{\eta} \mathbf{A}(\eta_0) \\ \nabla_{\eta} \mathbf{B}_{\mathcal{A}}(\eta_0) \end{bmatrix}$$

are linearly independent (LICQ). We assume strict complementarity when needed to ensure that the active set remains fixed.

Finally, the bordered KKT matrix is nonsingular and the second-order sufficient conditions hold on the critical cone. The former gives a locally unique differentiable KKT branch by the implicit function theorem, while the latter ensures that this branch consists of local minimisers. We perform the calculation at regular points with $\lambda_0 \neq 0$ and then take a one-sided limit as $\lambda_0 \rightarrow 0$. Nonsingularity is not required at the possibly unidentifiable exact-fit problem at $\lambda = 0$.

D.3 KKT perturbation

Define the Lagrangian

$$L(\eta, \mu, \nu; \lambda) = \mathcal{L}_{\text{ED}}(\eta; \hat{\mathbf{q}}) + \lambda a(\eta) + \mathbf{A}(\eta)^{\top} \mu + \mathbf{B}(\eta)^{\top} \nu.$$

At the reference point, the KKT conditions are

$$\begin{aligned} \nabla_{\eta} L(\eta_0, \mu_0, \nu_{0,\mathcal{A}}; \lambda_0) &= \mathbf{0}, \\ \mathbf{A}(\eta_0) &= \mathbf{0}, \\ \mathbf{B}_{\mathcal{A}}(\eta_0) &= \mathbf{0}, \\ \mathbf{B}_{\mathcal{A}^c}(\eta_0) &< \mathbf{0}, \\ \nu_{0,\mathcal{A}} &\geq \mathbf{0}, \\ \nu_{0,\mathcal{A}^c} &= \mathbf{0}. \end{aligned}$$

The third and sixth conditions, together with the active-set definition, imply complementary slackness. The perturbed point is

$$(\eta_1, \mu_1, \nu_{1,\mathcal{A}}; \lambda_1) = (\eta_0 + \delta\eta, \mu_0 + \delta\mu, \nu_{0,\mathcal{A}} + \delta\nu_{\mathcal{A}}; \lambda_0 + \delta\lambda).$$

Since the active set is fixed, its stationarity and feasibility conditions are

$$\begin{aligned}\nabla_{\eta} L(\eta_1, \mu_1, \nu_{1,\mathcal{A}}; \lambda_1) &= \mathbf{0}, \\ \mathbf{A}(\eta_1) &= \mathbf{0}, \\ \mathbf{B}_{\mathcal{A}}(\eta_1) &= \mathbf{0}.\end{aligned}$$

For compactness, define

$$\begin{aligned}\mathbf{p}_0 &= \mathbf{p}(\eta_0), & a_0 &= a(\eta_0), \\ \mathbf{A}_{\eta} &= \nabla_{\eta} \mathbf{A}(\eta_0), \\ \mathbf{B}_{\eta} &= \nabla_{\eta} \mathbf{B}_{\mathcal{A}}(\eta_0), \\ \mathbf{C}_0 &= \begin{bmatrix} \mathbf{A}_{\eta} \\ \mathbf{B}_{\eta} \end{bmatrix}, \\ \mathbf{J}_0 &= \nabla_{\eta} \mathbf{p}(\eta_0), \\ \mathbf{H}_0 &= \nabla_{\eta}^2 L(\eta_0, \mu_0, \nu_{0,\mathcal{A}}; \lambda_0) \\ &= -2\mathbf{J}_0^{\top} \mathbf{D} \mathbf{J}_0 \\ &\quad - 2 \sum_{k=1}^K [\mathbf{D}(\mathbf{p}_0 - \hat{\mathbf{q}})]_k \nabla_{\eta}^2 p_k(\eta_0) \\ &\quad + \lambda_0 \nabla_{\eta}^2 a(\eta_0) + \sum_r \mu_{0,r} \nabla_{\eta}^2 A_r(\eta_0) \\ &\quad + \sum_{r \in \mathcal{A}} \nu_{0,r} \nabla_{\eta}^2 B_r(\eta_0).\end{aligned}$$

Subtracting the reference-point stationarity condition from the perturbed stationarity condition and expanding to first order yields

$$\mathbf{0} = \mathbf{H}_0 \delta \eta + \mathbf{A}_{\eta}^{\top} \delta \mu + \mathbf{B}_{\eta}^{\top} \delta \nu_{\mathcal{A}} + \nabla_{\eta} a(\eta_0) \delta \lambda + \mathcal{O}(\delta \lambda^2).$$

Similarly, subtracting the reference feasibility conditions from their perturbed counterparts gives

$$\begin{aligned}\mathbf{0} &= \mathbf{A}_{\eta} \delta \eta + \mathcal{O}(\delta \lambda^2), \\ \mathbf{0} &= \mathbf{B}_{\eta} \delta \eta + \mathcal{O}(\delta \lambda^2).\end{aligned}$$

Writing $\delta \xi = (\delta \mu, \delta \nu_{\mathcal{A}})$ and stacking the stationarity and feasibility expansions gives

$$\begin{bmatrix} \mathbf{H}_0 & \mathbf{C}_0^{\top} \\ \mathbf{C}_0 & \mathbf{0} \end{bmatrix} \begin{bmatrix} \delta \eta \\ \delta \xi \end{bmatrix} = -\delta \lambda \begin{bmatrix} \nabla_{\eta} a(\eta_0) \\ \mathbf{0} \end{bmatrix} + \mathcal{O}(\delta \lambda^2).$$

Assuming this bordered KKT matrix is nonsingular, define

$$\begin{bmatrix} \mathbf{w}_{\eta} \\ \mathbf{w}_{\xi} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_0 & \mathbf{C}_0^{\top} \\ \mathbf{C}_0 & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \nabla_{\eta} a(\eta_0) \\ \mathbf{0} \end{bmatrix}.$$

In particular, the parameter component is

$$\mathbf{w}_{\eta} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{H}_0 & \mathbf{A}_{\eta}^{\top} & \mathbf{B}_{\eta}^{\top} \\ \mathbf{A}_{\eta} & \mathbf{0} & \mathbf{0} \\ \mathbf{B}_{\eta} & \mathbf{0} & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \nabla_{\eta} a(\eta_0) \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}.$$

It follows that

$$\begin{aligned}\delta \eta &= -\delta \lambda \mathbf{w}_{\eta} + \mathcal{O}(\delta \lambda^2), \\ \delta \xi &= -\delta \lambda \mathbf{w}_{\xi} + \mathcal{O}(\delta \lambda^2).\end{aligned}$$

Finally, using the leading-order Taylor expansions for $\mathbf{p}(\eta)$ and $a(\eta_1)$,

$$\begin{aligned}\mathbf{p}(\eta_1) &= \mathbf{p}_0 - \delta\lambda \mathbf{J}_0 \mathbf{w}_\eta + \mathcal{O}(\delta\lambda^2), \\ a(\eta_1) &= a_0 - \delta\lambda \nabla_\eta a(\eta_0)^\top \mathbf{w}_\eta + \mathcal{O}(\delta\lambda^2).\end{aligned}$$

When $\mathbf{p}_0 = \hat{\mathbf{q}}$,

$$\begin{aligned}\mathcal{L}_{\text{ED}}(\eta_1; \hat{\mathbf{q}}) &= -(\mathbf{p}(\eta_1) - \hat{\mathbf{q}})^\top \mathbf{D}(\mathbf{p}(\eta_1) - \hat{\mathbf{q}}) \\ &= -\delta\lambda^2 (\mathbf{J}_0 \mathbf{w}_\eta)^\top \mathbf{D}(\mathbf{J}_0 \mathbf{w}_\eta) + \mathcal{O}(\delta\lambda^3).\end{aligned}$$

There are no constant or $\mathcal{O}(\delta\lambda)$ terms at $\mathbf{p}_0 = \hat{\mathbf{q}}$. Since $\mathbf{D}$ is conditionally negative definite, the quadratic leading term is nonnegative.

For the one-sided branches in the main text, take regular reference points along $\eta_s(\lambda)$ and let $\lambda_0 \rightarrow 0$ from the corresponding side. Writing $\mathbf{J}_s(\lambda_0) = \nabla_\eta \mathbf{p}(\eta_s(\lambda_0))$ and letting $\mathbf{w}_{\eta,s}(\lambda_0)$ denote the KKT response for that branch, the KKT calculation gives

$$\left. \frac{d}{d\lambda} a(\eta_s(\lambda)) \right|_{\lambda=\lambda_0} = -\nabla_\eta a(\eta_s(\lambda_0))^\top \mathbf{w}_{\eta,s}(\lambda_0),$$

and

$$\left. \frac{d}{d\lambda} \mathbf{p}(\eta_s(\lambda)) \right|_{\lambda=\lambda_0} = -\mathbf{J}_s(\lambda_0) \mathbf{w}_{\eta,s}(\lambda_0).$$

Under the one-sided smoothness assumption in the main text, these induced quantities have one-sided limits at zero. Writing

$$c_s = \left. \frac{d}{d\lambda} a(\theta_s(\lambda)) \right|_{\lambda=0}, \quad \mathbf{v}_s = \left. \frac{d}{d\lambda} \mathbf{p}_{\theta_s(\lambda)} \right|_{\lambda=0},$$

the preceding expansions become

$$a(\theta_s(\lambda)) = a(\theta_s(0)) + c_s \lambda + \mathcal{O}(\lambda^2)$$

and

$$\mathcal{L}_{\text{ED}}(\theta_s(\lambda); \hat{\mathbf{q}}) = -\lambda^2 \mathbf{v}_s^\top \mathbf{D} \mathbf{v}_s + \mathcal{O}(|\lambda|^3).$$

Since $\mathbf{v}_s$ has zero sum, conditional negative definiteness of $\mathbf{D}$ gives $-\mathbf{v}_s^\top \mathbf{D} \mathbf{v}_s \geq 0$. This is the local extrapolation result used in the main text. The KKT calculation does not by itself establish that the branch is globally optimal; the global-minimiser assumption is what is needed for the bound-recovery corollary.

E Analytical Bounds

In this section, we give analytical bounds for the Mediated graph, $Z \rightarrow T \leftrightarrow M \rightarrow Y$. Here $\leftrightarrow$ denotes mutual confounding between T and M . All of Z, T, M, Y are binary, there is no restriction on the confounding variable. This graph structure is beyond the assumptions required by CausalOptim.

Let the ATE from intervening on node A with target node B be $\text{ATE}_{A \rightarrow B}$.

Theorem 2. *Let $\delta_Y = \Pr(Y = 1 | M = 1) - \Pr(Y = 1 | M = 0)$. Then $\text{ATE}_{T \rightarrow Y} = \delta_Y \text{ATE}_{T \rightarrow M}$.*

Proof. Because the effect of T on Y is fully mediated by M , and the effect of M on Y is unconfounded, the mediation formula gives

$$\begin{aligned}\Pr(Y = 1 \mid \text{do}(T = t)) &= \\ \sum_{m \in \{0,1\}} \Pr(Y = 1 \mid M = m) \Pr(M = m \mid \text{do}(T = t)).\end{aligned}$$

Taking the difference between $t = 1$ and $t = 0$ yields

$$\begin{aligned} \text{ATE}_{T \rightarrow Y} &= \delta_Y \left[\Pr(M = 1 \mid \text{do}(T = 1)) \right. \\ &\quad \left. - \Pr(M = 1 \mid \text{do}(T = 0)) \right] \\ &= \delta_Y \text{ATE}_{T \rightarrow M}. \end{aligned}$$

where the final equality follows because M is binary. $\square$

Thus, in order to compute bounds of $\text{ATE}_{T \rightarrow Y}$, we only need the bounds of $\text{ATE}_{T \rightarrow M}$. This is exactly the binary IV case which is known (Balke and Pearl, 1997).

F Synthetic Data Generation

Given a causal directed acyclic graph (DAG), we construct a structural equation model (SEM) and generate its nodes in topological order. Let $\mathcal{V}$ denote the set of endogenous nodes in the graph and $\mathcal{U}$ the set of exogenous variables. For each node X_v , $v \in \mathcal{V}$, we distinguish between its endogenous parents

$$\text{pa}_{\mathcal{V}}(v) \subseteq \mathcal{V}$$

and its exogenous parents

$$\text{pa}_{\mathcal{U}}(v) \subseteq \mathcal{U}.$$

Endogenous parents correspond to other variables represented explicitly in the causal graph, whereas exogenous parents represent latent sources of dependence, such as unobserved confounders.

The exogenous variables and node-specific noise terms are sampled as

$$\begin{aligned} U_j &\sim \text{Uniform}(0, 1), & j &\in \mathcal{U}, \\ \varepsilon_v &\sim \text{Uniform}(0, 1), & v &\in \mathcal{V}, \end{aligned}$$

with the U_j 's and ε_v 's mutually independent unless the graph explicitly specifies a shared exogenous parent. Each endogenous node is then generated according to the structural assignment

$$X_v = f_v(X_{\text{pa}_{\mathcal{V}}(v)}, U_{\text{pa}_{\mathcal{U}}(v)}, \varepsilon_v), \quad v \in \mathcal{V},$$

where $X_{\text{pa}_{\mathcal{V}}(v)}$ and $U_{\text{pa}_{\mathcal{U}}(v)}$ denote the collections of endogenous and exogenous parent values, respectively. A shared exogenous parent may enter the structural assignments of several nodes, thereby inducing latent dependence between them.

F.1 Random structural functions

The nonlinear structural mechanisms are sampled once when a dataset instance is created. They are represented by random Fourier functions mapping $[0, 1]$ to $(0, 1)$. For a scalar input x , a function is defined as

$$\begin{aligned} g(x) &= b + \sum_{k=1}^K \left[a_k \cos\left(2\pi\left(k + \frac{1}{2}\right)x\right) + c_k \sin\left(2\pi\left(k + \frac{1}{2}\right)x\right) \right], \\ r(x) &= \text{sigmoid}(\alpha g(x)). \end{aligned}$$

where we have $K = 10$ Fourier components and $\alpha = 2$. The coefficients are independently sampled as

$$b \sim \mathcal{N}(0, 1), \quad a_k, c_k \sim \mathcal{N}(0, k^{-2\delta}), \quad \delta = 1.5.$$

Thus, the standard deviation of the coefficient at frequency k is $k^{-\delta}$. This decay suppresses high-frequency components and produces smooth but nonlinear functions. Half-integer frequencies are used to avoid forcing the function to have equal values at the two boundaries of the unit interval.

A separate random function is drawn for each relevant configuration of the discrete parents. For example, with binary Z , the treatment probability is

$$p_D(z, u) = \begin{cases} r_{D,0}(u), & z = 0, \\ r_{D,1}(u), & z = 1, \end{cases}$$

and the binary treatment is generated by inverse-transform sampling,

$$D = \mathbf{1}\{\varepsilon_D < p_D(Z, U)\}.$$

For a multi-valued parent, such as a ternary instrument, one independent function is constructed for each parent state and the function corresponding to the realised state is selected.

For a categorical node with $k > 2$ states, the generator constructs k random scores $s_0, \dots, s_{k-1}$. In the ternary-outcome experiments, it first obtains a treatment-specific scalar representation,

$$h(D, U) = \begin{cases} r_{h,0}(U), & D = 0, \\ r_{h,1}(U), & D = 1, \end{cases}$$

and evaluates the category scores at this value. The conditional probabilities are

$$\pi_j(D, U) = \frac{\exp\{s_j(h(D, U))\}}{\sum_{\ell=0}^{k-1} \exp\{s_\ell(h(D, U))\}}, \quad j = 0, \dots, k-1.$$

The realized category is then obtained using the cumulative probabilities:

$$Y = \min \left\{ j : \varepsilon_Y \leq \sum_{\ell=0}^j \pi_\ell(D, U) \right\}.$$

The Fourier coefficients determine the structural mechanisms and remain fixed for all observations in one dataset. Changing the dataset seed draws a new collection of functions and therefore a new SEM. Conditional on these fixed mechanisms, individual observations differ because the root variables and exogenous noise variables are resampled.

F.2 Observational samples

An observational dataset of size b is obtained by repeatedly sampling the root and noise variables and then evaluating all structural assignments in topological order.

Since the problem is only partially identified, we do not use the ATE of the SEM used to generate the dataset. There are many other SEMs with identical observational distributions. Rather, we use the observational distribution to compute bounds on the ATE. This is done by monte-carlo sampling for $\mathbf{q}$ over a large number of observations, then $\mathbf{q}$ is used in the analytic solver for the true bounds.

F.3 Graph details

For each graph, the U node is unobserved, acting as unobserved confounding. In the Binary IV, Proxy and Mediated graphs, all observed variables are binary. In the Ternary IV graph, the treatment remains binary (so ATE is easily calculated) while node Z and node Y are both ternary. The datasets are generated using a random seeds from 0 to 9, with the exception of the Mediated dataset. Many datasets from this graph gave nearly 0 ATE. Instead, for this graph, the first 10 datasets to give nontrivial ATE values were used. The final seeds were $\{1, 3, 5, 6, 7, 10, 11, 16, 17\}$.

G Analysis of the baselines

In Section 5.4, we found that the baseline methods did not scale with model size. Specifically, the model performance degraded as the neural network widths increased, even with μP learning rate scaling. In this section, we analyse the cause of this failure.

Width	Lower error	Upper error
16	0.008	0.010
32	0.010	0.011
64	0.009	0.014
128	0.062	0.151
256	0.371	0.399
512	0.572	0.780
1024	0.855	0.931

Table 2: Error of the GAN generator from BCCE for different NN widths for the upper bound model and lower bound model. Lower error indicates better fit. The learning rate follows μP scaling.

Width	α	$\chi_{\min}$	$\chi_{\max}$
16	0.131	12.48	10.55
32	0.132	22.54	24.51
64	0.136	14.18	27.61
128	0.136	0.21	36.56
256	0.128	7.62	20.97
512	0.137	8.07	29.98
1024	0.135	20.15	18.58

Table 3: Final parameters of the NeuralPartialID models for different width NNs. $\chi_{\min}$, and $\chi_{\max}$ are the final Lagrange multipliers, for the lower and upper bounds and α is the fit quality threshold.

BCCE is a GAN architecture, where a generator and discriminator are adversarially optimised. As the NN widths were scaled up, we found an instability where the discriminator becomes too strong and no useful gradient feedback is given to the generator. This can be seen by the high generator error as the width of the NNs becomes large, as seen in Table 2 where the model is evaluated on a ternary IV example. The error measured is the largest absolute difference in probability between the predicted and true distributions. This failure mode is well documented and techniques to enhance their stability have been developed (Salimans et al., 2016), but is undesirable for the user. As a result of this issue, BCCE struggles on more complex tasks.

Similarly, NeuralPartialID’s performance slightly degrades with larger NN widths. This method is based on a Lagrange multiplier method, shared with (Balazadeh Meresht et al., 2022) that solves the minimax optimisation problem,

$$\min_{\theta} \max_{\chi \geq 0} \text{ATE}_{\theta} + \chi (S(P_{\theta}, P) - \alpha)$$

where $S(P_{\theta}, P)$ denotes the Sinkhorn distance (Cuturi, 2013) between the predicted and observed distributions. The model parameters θ are updated by gradient descent, while the Lagrange multiplier (χ) is updated according to $\chi \leftarrow \chi + \chi_{\text{LR}} [S(P_{\theta}, P) - \alpha]$. The fit quality threshold α is derived from a pretraining phase where the average treatment effect is not optimised.

In principle, the exact constrained solution is recovered in the limiting case $\chi \rightarrow \infty$ and $\alpha = 0$. In practice, however, the update rate of χ must be carefully balanced against the convergence rate of the model parameters. If these rates are poorly matched, training may become prohibitively slow or unstable. Moreover, the resulting bounds are highly sensitive to the choice of α . In practice, the algorithm converges to $\alpha > 0$ and the constraint is relaxed, which tends to produce bounds that are wider than the theoretically optimal bounds. Table 3 displays the fitted α and final χ values for different width models, on the same IV ternary example, where we see α is always around 0.13 and χ is unstable and fluctuates as NN width is increased. The desired scaling of $\chi \rightarrow \infty$ does not occur.